1. Write a shell script which will generate the O/P as follows

\*

\*\*

\*\*\*

\*\*\*\*

[admin@hostname01 ~]$ nano star\_patt.sh

* for((i=1; i<=4; i++))
* do
* for((j=1; j<=i; j++))
* do
* echo -n "\*"
* done
* echo
* done
* [admin@hostname01 ~]$ ./star\_patt.sh
* bash: ./star\_patt.sh: Permission denied
* [admin@hostname01 ~]$ chmod +x star\_patt.sh
* [admin@hostname01 ~]$ ./star\_patt.sh
* \*
* \*\*
* \*\*\*
* \*\*\*\*
* [admin@hostname01 ~]$

1. Accept the first name, middle name, and last name of a person in variables fname, mname and lname respectively. Greet the person (take his full name) using appropriate message.

* [admin@hostname01 ~]$ nano greet.sh
* read -p "Enter your first name please:" fname
* read -p "Enter your middle name please:" mname
* read -p "Enter your last name please:" lname
* echo "Hello,$fname $mname $lname!! Welcome to world!!!"
* [admin@hostname01 ~]$ chmod +x greet.sh
* [admin@hostname01 ~]$ ./greet.sh
* Enter your first name please: Shivraj
* Enter your middle name please: Shantesh
* Enter your last name please: Kapase
* Hello,Shivraj Shantesh Kapase!! Welcome to world!!!

1. Display the name of files in the current directory along with the names of files with maximum & minimum size. The file size is considered in bytes.

🡪 admin@hostname01 ~]$ nano filesize.sh

#!/bin/bash

find . -maxdepth 1 -type f ! -name "dev" ! -name "null" -exec ls -l {} \; | >

max=$(tail -1 temp.txt)

min=$(head -1 temp.txt)

echo "Files in current directory:"

cat temp.txt

echo "File with maximum size: $max"

echo "File with minimum size: $min"

rm temp.txt

[admin@hostname01 ~]$ ./filesize.sh

Files in current directory:

./temp.txt 0

./dept.lst 13

./desig.lst 13

./.bash\_logout 18

./.lesshst 20

./demofile 23

./friends 25

./newfriend 25

./users 25

./networrk.txt 523

./dir 1086

./.bash\_history 7299

./.viminfo 8292

./.networkk.txt.swp 12288

./.network.txt.swp 12288

File with maximum size: ./.network.txt.swp 12288

File with minimum size: ./temp.txt 0

1. Write a script which when executed checks out whether it is a working day or not?

(Note: Working day Mon-Fri)

* admin@hostname01 ~]$ nano weather.sh
* #!/bin/bash
* # user input for the day
* read -p "Enter a day of the week which you want to check: " day
* # Convert the input to lowercase for case-insensitive comparison
* day=$(echo "$day" | tr '[:upper:]' '[:lower:]')
* if echo "$day" | grep -iqE "^(mon|tue|wed|thu|fri)$"; then
* echo "$day is a working day."
* else
* echo "$day is not a working day."
* Fi
* [admin@hostname01 ~]$ chmod +x weather.sh
* [admin@hostname01 ~]$ ./weather.sh
* Enter a day of the week which you want to check: mon
* mon is a working day.

1. Write a script that accepts a member into HP health club, if the weight of the person is withing the range of 30-250 Kgs.

🡪 [admin@hostname01 ~]$ nano hp.sh

!/bin/bash

read -p "Enter your weight in kg: " weight

if [ "$weight" -lt 30 ] || [ "$weight" -gt 250 ]; then

echo "Sorry, your weight is outside the acceptable range (30-250 kg). You >

else

echo "Welcome to HP Health Club! You have been successfully accepted."

fi

[admin@hostname01 ~]$ chmod +x hp.sh

[admin@hostname01 ~]$ ./hp.sh

Enter your weight in kg: 59

Welcome to HP Health Club! You have been successfully accepted.

1. Write a shell script that greets the user with an appropriate message depending on the system time.

🡪 [admin@hostname01 ~]$ nano timegreet.sh

#!/bin/bash

read -p "Enter the hour (0-23 only): " hour

if [ "$hour" -lt 0 ] || [ "$hour" -ge 24 ]; then

echo "Invalid input! Please enter a number between 0 and 23."

exit 1

fi

# Greet based on the time of day

if [ "$hour" -ge 5 ] && [ "$hour" -lt 12 ]; then

echo "Good Morning!"

elif [ "$hour" -ge 12 ] && [ "$hour" -lt 17 ]; then

echo "Good Afternoon!"

elif [ "$hour" -ge 17 ] && [ "$hour" -lt 21 ]; then

echo "Good Evening!"

else

echo "Good Night!"

fi

[admin@hostname01 ~]$ chmod +x timegreet.sh

[admin@hostname01 ~]$ ./timegreet.sh

Enter the hour (0-23): 22

Good Night!

1. A data file file has some student records including rollno, names and subject marks. The fields are separated by a “:”. Write a shell script that accepts roll number from the user, searches it in the file and if the roll number is present - allows the user to modify name and marks in 3 subjects.   
   If the roll number is not present, display a message “Roll No Not Found”. Allow the user to modify one record at a time.

🡪[admin@hostname01 ~]$ nano modfify\_stu.sh

#!/bin/bash

file="student\_records.txt"

read -p "Enter the roll number to search: " rollno

record=$(grep "^$rollno:" "$file")

if [ -n "$record" ]; then

echo "Record found: $record"

IFS=":" read -r roll name marks1 marks2 marks3 <<< "$record"

# Allow user to modify name and marks for modification

read -p "Enter new name (current: $name): " new\_name

read -p "Enter new mark for subject 1 (current: $marks1): " new\_marks1

read -p "Enter new mark for subject 2 (current: $marks2): " new\_marks2

read -p "Enter new mark for subject 3 (current: $marks3): " new\_marks3

sed -i "s/^$rollno:$name:$marks1:$marks2:$marks3$/$rollno:$new\_name:$new\_marks1:$new\_marks2:$new\_marks3/" "$file"

echo "Record updated successfully!"

else

# If roll number is not found

echo "Roll No Not Found"

fi

[admin@hostname01 ~]$ ./modfify\_stu.sh

Enter the roll number to search: 102

Record found: 102:Supreet:85:90:95

Enter new name (current: Supreet): Om

Enter new mark for subject 1 (current: 85): 90

Enter new mark for subject 2 (current: 90): 95

Enter new mark for subject 3 (current: 95): 97

Record updated successfully!

[admin@hostname01 ~]$ cat studentrecord.txt

101:Shivraj:80:90:70

102:Ash:90:95:97

103:Rohan:60:70:80

1. Modify program 7 to accept the RollNo from the command line.

🡪 [admin@hostname01 ~]$ nano modfify\_stu.sh

# To Accept roll number

read -p "Enter the roll number to search: " rollno

[admin@hostname01 ~]$ ./modfify\_stu.sh

Enter the roll number to search: 102

Record found: 102:Supreet:85:90:95

1. Modify the program 7 to accept the RollNo and display the record and ask for delete confirmation. Once confirmed delete the record and update the data file.

🡪 [admin@hostname01 ~]$ nano del\_stu.sh

#!/bin/bash

file="studentrecord.txt"

if [ -z "$1" ]; then

read -p "Enter the roll number to search: " rollno

else

# Use the command line argument for roll number

rollno=$1

fi

record=$(grep "^$rollno:" "$file")

if [ -n "$record" ]; then

echo "Record found: $record"

read -p "want to delete record type y : " confirm

if [ "$confirm" == "y" ] || [ "$confirm" == "Y" ]; then

sed -i "/^$rollno:/d" "$file"

echo "Record deleted successfully!"

else

echo "Deletion aborted."

fi

else

echo "Roll No Not Found"

fi

[admin@hostname01 ~]$ chmod +x del\_stu.sh

[admin@hostname01 ~]$ ./del\_stu.sh

Enter the roll number to search: 103

Record found: 103:Sushil:60:70:80

want to delete record type y : y

Record deleted successfully!

1. Write a script that takes a command line argument and reports on its file type (regular file, directory file, etc.). For more than one argument generate error message.

🡪 [admin@hostname01 ~]$ nano filereport.sh

if [ "$#" -ne 1 ]; then

echo "Error: Please provide exactly one argument."

echo "Usage: $0 <file\_path>"

exit 1

fi

file\_path=$1

if [ -e "$file\_path" ]; then

file\_type=$(file "$file\_path")

echo "$file\_type"

else

echo "Error: $file\_path does not exist."

exit 1

fi

[admin@hostname01 ~]$ chmod +x filereport.sh

[admin@hostname01 ~]$ ./filereport.sh networrk.txt

networrk.txt: ASCII text, with very long lines

[admin@hostname01 ~]$ ./filereport.sh errorfile.txt

Error: errorfile.txt does not exist.

[admin@hostname01 ~]$ ./filereport.sh styles

styles: directory

1. Add some student records in the “student” file manually. The fields to be considered are “RollNo”, “Name”, “Marks\_Hindi”, “Marks\_Maths”, “Marks\_Physics”.

🡪 [admin@hostname01 ~]$ touch student

[admin@hostname01 ~]$ vim student

[admin@hostname01 ~]$ cat student

101:Rohan:70:85:92

102:Riya:70:75:80

103:Rohan:90:68:85

[admin@hostname01 ~]$ nano studentrec.sh

[admin@hostname01 ~]$ chmod +x studentrec.sh

[admin@hostname01 ~]$ ./studentrec.sh

Enter Roll No: 104

Enter Name: Om

Enter Marks in Hindi: 78

Enter Marks in Maths: 90

Enter Marks in Physics: 89

Total Marks: 257

Percentage: 85%

Write a script which does the following:

* 1. If the roll number already exists, then store the record and the following message   
     “roll number exists” in a log file “log1”.

🡪 [admin@hostname01 ~]$ ./studentrec.sh

Enter Roll No: 102

Enter Name: Sup

Roll number 102 exists.

* 1. If the marks in the subjects is not in the range of 1 – 99 then store such a record followed by a message “marks out of range” in “log1”

🡪 [admin@hostname01 ~]$ ./studentrec.sh

Enter Roll No: 106

Enter Name: Priti

Enter Marks in Hindi: 90

Enter Marks in Maths: 78

Enter Marks in Physics: 101

Marks out of range.

* 1. If the data is valid, the calculate total, percentage, grade and display on the terminal

🡪

[admin@hostname01 ~]$ ./studentrec.sh

Enter Roll No: 105

Enter Name: Sai

Enter Marks in Hindi: 90

Enter Marks in Maths: 34

Enter Marks in Physics: 67

Total Marks: 191

Percentage: 63%

**Below is the shell code for this** : 🡪

#!/bin/bash

student\_file="student"

log\_file="log1"

read -p "Enter Roll No: " roll\_no

read -p "Enter Name: " name

read -p "Enter Marks in Hindi: " marks\_hindi

read -p "Enter Marks in Maths: " marks\_maths

read -p "Enter Marks in Physics: " marks\_physics

#**Here I checked for roll number already exists or not**

if grep -q "^$roll\_no:" "$student\_file"; then

echo "Roll number $roll\_no exists."

echo "roll number exists" >> "$log\_file"

exit 1

fi

#**Here I checked for its in range or not**

if [ "$marks\_hindi" -lt 1 ] || [ "$marks\_hindi" -gt 99 ] || [ "$marks\_maths" -lt 1 ] || [ "$marks\_maths" -gt 99 ] || [ "$marks\_physics" -lt 1 ] || [ "$marks\_physics" -gt 99 ]; then

echo "Marks out of range."

echo "marks out of range" >> "$log\_file"

exit 1

fi

echo "$roll\_no:$name:$marks\_hindi:$marks\_maths:$marks\_physics" >> "$student\_file"

total=$((marks\_hindi + marks\_maths + marks\_physics))

percentage=$((total / 3))

echo "Total Marks: $total"

echo "Percentage: $percentage%"

Verify Email Address :

[admin@hostname01 ~]$ nano gmail.sh

#!/bin/bash

read -p "Enter your email address: " email

email\_patt="^[a-zA-Z0-9]+@[a-zA-Z0-9]+\.(com|in)$"

if [[ $email =~ $email\_patt ]]; then

echo "The email address is valid."

else

echo "Invalid email address."

fi

[admin@hostname01 ~]$ chmod +x gmail.sh

[admin@hostname01 ~]$ ./gmail.sh

Enter your email address: shivraj@gmail.com

The email address is valid.

[admin@hostname01 ~]$ ./gmail.sh

Enter your email address: opadads1@yahoo.edu

Invalid email address.

To check the Mobile number Start withg +91 |(90/89 start with)|

[admin@hostname01 ~]$ nano mobilechk.sh

[admin@hostname01 ~]$ chmod +x mobilechk.sh

[admin@hostname01 ~]$ ./mobilechk.sh

Enter your mobile number: 7709913698

Invalid mobile number. Ensure it starts with +91 and begins with 89 or 90.

[admin@hostname01 ~]$ ./mobilechk.sh

Enter your mobile number: +919056782390

The mobile number is valid.

[admin@hostname01 ~]$

#!/bin/bash

read -p "Enter your mobile number: " mobile\_number

mobile\_patt="^\+91(89|90)[0-9]{8}$"

if [[ $mobile\_number =~ $mobile\_patt ]]; then

echo "The mobile number is valid."

else

echo "Invalid mobile number. Ensure it starts with +91 and begins with 89 or 90."

fi